Q1 Team Name

0 Points

INSYNC

Q2 Commands

5 Points

List the commands used in the game to reach the ciphertext.

go -> wave -> dive -> go -> read -> password

Q3 Analysis

50 Points

Give a detailed description of the cryptanalysis used to figure out the password. (Use Latex wherever required. If your solution is not readable, you will lose marks. If necessary the file upload option in this question must be used TO SHARE IMAGES ONLY.)

#Step-1:
On the first screen of game we got hint like "The passage seems to be leading to some deep underground well! " therefore we used go command , then we got another hint on next screen "Desperate, you try to grab something, but there is nothing to grab ..." therefore we used $wave$,then we got another hint on next screen "Swimming in opposite direction yields the same result!" therefore we used $dive$ command and after that we got another hint on next screen "A well-lit passage is carved in the wall, and seems to be leading somewhere deep inside..." therefore we used go command,then we got another hint in next screen "You go around the hall, exploring its details. The marble is is
very good quality. (How did someone get all this marble here, you wonder?)" therefore we used $read$ command. After this, we got detailed

instructions on how to decrypt the cipher which are mentioned below :
(1) We have given a block of size 8 bytes as $8 * 1$ vector over F_{128} constructed using the degree 7 irreducible polynomial $x^7 + x + 1$ over F_2.
(2) There are two transformations mainly :
->F_{128} was used to define linear transformation that is deduced using invertible $8 * 8$ key matrix A.
->E is a $8 * 1$ vector expresses exponentiation whose elements are in range (1 - 126).
(3)We have to apply these transformations in the sequence $EAEAE$ on the input block to obtain the output block. It was mentioned that both E and A are part of the key. E is applied on a block by taking the i^{th} element of the block and raising it to the power given by i^{th} element in E.

#We got our cipher text: **"ijlnglfufhlfmhfojlhilriklnlpmuik"** , using command $password$

#Step-2:
-> After obtaining cipher text, the game was designed in such a manner that we can input plaintexts and get the corresponding cipher texts. We used this feature and analyzed the cipher texts which we were getting and noticed that the range of cipher text is limited by a specific range which was (f-u), that is there were only 16 alphabets in range of cipher text.
->we mapped these 16 alphabets in binary representation as follows:
f: 0000 g:0001 h:0010 i:0011
j: 0100 k:0101 l:0110 m:0111
n:1000 o:1001 p:1010 q:1011
r: 1100 s:1101 t:1110 u:1111

->As we were provided with instructions on magical screen that each byte of block has a range (0-127) because it is defined over F_{128}. Conclusion that we made out of it was that every byte of the input is from 'ff' to 'mu' has been mapped to 0 to 127 and signed bit or the msb of each byte is set as 0 . After further analysis, we saw that padding of 'ff' was incorporated. So, $'jk'$ after padding will become $'jkfffffffffffffff'$

#Step-3:
->To carry out decryption, we started by generating plaintext in format

$(C^{8-i} * P * C^{i-1})$ where $(1 \leq i \leq 8)$ and $C = 0$, P has range $(1 - 127)$ over F_{128}

->Initially we tried to extract 1^{st} diagonal of A and complete E

->'generatePairs.py' was used to generate plain texts.

->The same code 'generatePairs.py' was used to generate the corresponding cipher texts by automating the task using paramiko.

-> Ciphertext which we obtained after EAEAE transformation was :

$C_i = (A_{ii} * (A_{ii} * P_i^{E_i})^{E_i})^{E_i}$

note the abbreviations used :

$C_i \implies$ Output block's i^{th} byte ,$P_i \implies$ Plain Text block's i^{th} byte, $A_{ii} \implies A[i][i]$ belongs to Linear Transformation Matrix, $E_i \implies E[i]$ belongs to Exponentiation Vector.

#Step-4:

Now we tried all possible combinations of A_{ii} and E_i, because we know that the valid range of A and E. We figured out in previous stages that A_{ii} will have values in the range (0 -127) and E_i has valid range between (1 - 126). Exponentiation and Multiplication are performed over F_{128} by irreducible polynomial $x^7 + x + 1$. Addition operation over F_{128} is performed using bit-wise XOR.

->First we computed C_i for each of combination for every P_i . After this computation, we checked C_i with i^{th} byte of cipher text. If matches then A_{ii} is the valid candidate for i^{th} diagonal value and E_i is possible i^{th} element of E. After executing over all plaintext and ciphertext pairs, we obtained three possible values for A_{ii} and E_i for every i^{th} byte.

->Values we obtained for A are as follows :

$A_{00} : [84, 28, 113], A_{11} : [120, 70, 104], A_{22} : [43, 14, 72], A_{33} : [33, 12, 122], A_{44} : [73, 112, 57], A_{55} : [100, 11, 122], A_{66} : [27, 20, 124], A_{77} : [38, 35, 39]$

->Values we obtained for E is as follows :

$E_{00} : [18, 21, 88], E_{11} : [26, 113, 115], E_{22} : [40, 89, 125], E_{33} : [64, 73, 117], E_{44} : [45, 93, 116], E_{55} : [23, 48, 56], E_{66} : [26, 113, 115], E_{77} : [18, 21, 88]$

#Step-5:

Now we have to find non-diagonal elements A and get correct values out of three pairs which we got in previous step. We used the equation $C_i = (A_{ii} * (A_{ii} * {P_i}^{E_i})^{E_i})^{E_i}$ and additional (plaintext,ciphertext) pairs and then executed on pairs which we obtained in previous stage to find elements in range (0 - 127) which satisfied equation mentioned above. Valid $A_{i,j}$ was calculated with the help of $A_{i,i}$ and $A_{j,j}$ in a triangular manner. While finding values for non - diagonal elements of $A_{i,i}$, we noticed that there were two values for $A_{0,7}$(i.e 1st row, last element) which were $(68, 122)$. Both of these values were possible, but we decided to go with value 68 because it resulted in the correct decryption of our ciphertext(which was padded with six zeros). But for all other elements, we got unique valid candidates.

#Step-6:
Using the above steps, we reduced the search space for possible candidates for key pairs because there were very few pairs using which we can find the value of next elements of diagonal element.
Finally we extracted the complete A matrix by iterating over all the available values using E vector, the previously computed matrix A, and $C_i = (A_{ii} * (A_{ii} * {P_i}^{E_i})^{E_i})^{E_i}$.

->Final A Matrix :
$[[84, 124, 19, 98, 101, 29, 16, 68],$
$[0, 70, 28, 17, 38, 53, 119, 9],$
$[0, 0, 43, 4, 1, 28, 9, 82],$
$[0, 0, 0, 12, 104, 47, 101, 27],$
$[0, 0, 0, 0, 112, 100, 29, 10],$
$[0, 0, 0, 0, 0, 11, 89, 70],$
$[0, 0, 0, 0, 0, 0, 27, 25],$
$[0, 0, 0, 0, 0, 0, 0, 38]]$
->Final E Vector :
$[18, 113, 40, 73, 93, 48, 26, 18]$

#Step-7:
Our ciphertext was = $ijlnglfufhlfmhfojlhilriklnlpmuik$. We broke down it into two parts each of 8 byte each because our block size is 8bytes. We took one part and for each byte, we iterated over every value of byte $(0 - 127)$and performed $EAEAE$ transformations and checked

if it matches with corresponding byte of password. If yes, we accept it as valid and processed further upcoming bytes of that plaintext block. Same procedure was applied for the second part. We got the final plaintext by joining both the parts which was:
$rrqoeusofo000000$. The above process was done using the code "analyzeDecrypt.py".

Finally, we removed the padding of zeros to get final password as $rrqoeusofo$.

▾ Cryptology level_5 INSYNC.pdf Download

Q4 Password

5 Points

What was the final commands used to clear this level?

rrqoeusofo

Q5 Codes

0 Points

It is mandatory that you upload the codes used in the cryptanalysis. If you fails to do so, you will be given 0 for the entire assignment.

▾ INSYNC.zip Download

```
1  Binary file hidden. You can download it using the button above.
```

Assignment 5

GRADED

2 DAYS, 23 HOURS LATE

GROUP

Punit Chaudhari
Aman Mittal
Piyush Gangle
View or edit group

TOTAL POINTS

50 / 60 pts

QUESTION 1

Team Name **0** / 0 pts

QUESTION 2

Commands **5** / 5 pts

QUESTION 3

Analysis R **40** / 50 pts

QUESTION 4

Password **5** / 5 pts

QUESTION 5

Codes **0** / 0 pts